Linux: CLI (command line interface) Commands

# INTRODUCTION:

Earlier, there was no UI for devices hence commands had to be given in order to perform any activity. But Now many devices doesn’t need GUI since there is no need for user interaction (Ex: server, backend device etc) hence CLI is still in use.

Shell 🡪 A software program that accepts commands from Terminal, processes it and shares the result back on terminal.

Terminal 🡪 A program that runs shell. (Mainly a UI interface between Shell and user)

Bash 🡪 is a shell and command language written for Interface. (bash is standard shell in Linux, Unix machines).

Refer: <http://www.sanfoundry.com/1000-linux-command-tutorials/> for shell commands.

Shell commands

* Functions that run & perform activity
* Case sensitive (all command are lowercase).

# **SHELL COMMANDS:**

|  |  |
| --- | --- |
| **Command** | **Description with usage** |
| ls | Lists all the files |
| $ ls ; displays all the file  $ ls \*jpg ; displays all files ending with jpg.  $ ls \* JPG; displays all files ending with JPG.  $ ls \*(jpg, png,mpeg) ; displays all files ending with jpg/png/mpeg.  $ ls \*(jpg, png): == ls \*jpg, \*png ; == ls \*{jp,pn}g ; (all display same o/p)  $ ls \*j?g: (? 🡪 used for one character matching, so all files ending with j(any character)g)  $ ls –a : displays hidden file as well.  $ ls –lh: displays files with their size  $ ls –lah: displays all files (hidden as well) with their sizes. |
| Conditions on naming a File/directory:   * It can contain any character but cannot have ‘/’ * If name starts with any special character (such as #$% etc) put the name in ‘ ‘ (called as quoting) or precede each special character with ‘/’ (called as escaping).   Ex: ?$file == ‘?$file’ == /?/$file | |
| mkdir XXX | To create a folder/directory. |
| $ Mkdir file\_name; 🡪creates a directory file\_name;  $ mkdir a.txt ; 🡪 creates a folder with name “a.txt” not a txt file. So to create a file with proper format use “touch” command. |
| rmdir XXX | To remove any folder/directory. |
| Ex: $ rmdir file\_name; |
| rm XXX | Remove any file;  Ex: $ rm file.txt 🡪 removes file from the directory. |
| touch XXX | To create any file. |
| Ex: $ Touch a.txt; 🡪 creates a text file  $ Touch a.xlsx; 🡪 creates an excel file. |
| clear | Clears the CLI of all activity/commands. |
| find | Displays all the files(hidden as well) with their path (similar to ls command but shows file path instead of just file name) |
| where XXX /  whereis XXX | To check where a particular command exists.  Ex: $ where ls; 🡪 o/p = C:\Program Files\Git\user\bin\ls.exe |
| echo XX | Prints data on console/terminal.  $ echo Hi\n 🡪 Hi\n is displayed on console  $ echo hello! Its me>>file.txt 🡪 add data (hello! Its me) into the file.txt |
| printf XX | Same as Echo but it considers \n as line break.  Ex: $ printf “hey \n hello” 🡪 output will be as below:  hey  hello |
| Consider below file structure for ‘cd’ command reference:  (parent directory) Home 🡪 Supriya (base directory) 🡪 file1 🡪document2🡪 w.txt;  🡪 user1 🡪 project1, e.txt, hello.docx  🡪 user 2 🡪 welcome, how.pdf, project2 | |
| cd | Change directory, move to base/working directory of user from anywhere in the filesystem. |
|  | Ex: your CD = document2; 🡪 $ cd ; 🡪 your CD = Supriya;  Ex: your CD = project2; 🡪 $ cd ; 🡪 your CD = user2;  Ex: your CD= home; 🡪 $ cd ; 🡪 your CD = Supriya (base directory) |
| cd [path] | Move to the path specified only if it exists. |
|  | Your CD = file1; 🡪 $ cd document2; 🡪 moves to document2 folder.  Your CD = project1; 🡪 $ cd p ; 🡪 ERROR, no file found  Your CD= home; 🡪 $ cd Supriya/file1/document2/ ; 🡪 moves to diocument2 folder. |
| cd - | To switch between last 2 recently used directories. |
| cd .. | Move to home directory (only from home directory+1 stage).  cd .. = cd /home  cd ../Supriya = cd /home/Supriya |
|  | Ex: your CD = project1; 🡪 $ cd .. ; 🡪 your CD = user1;  Ex: your CD = user2 ; 🡪 $ cd .. ; 🡪 your CD = home; |
| exit | To logout from the machine |
| Ctrl + D | To logout from the machine |
| quit | To exit the terminal window without closing shell. |
|  | Ex: $ bc 🡪 displays license information, to come out of window use $ quit. |
| ‘ “ ( ) | To get back shell prompt type same character you typed as shown below.  ‘ – ‘ ;  “ – “ ;  ( - ) ; |
| date | Displays date and time w.r.t country |
|  | $date ; displays Thu Sep 28 09:00:01 IST 2017 |
| hostname | Displays the name of the host i.e., name logged in as. |
| host [URL] | Provides information of the URL hosted.  Ex: $ host [www.google.co.in](http://www.google.co.in)   * Shows IP addresses hosted by google |
| echo xx | Displays any text followed by echo in the shell screen |
|  | $ echo welcome home; |
| expr xx | Expression commands to do any arithmetic/logical function. |
|  | $ expr 2 + 2 ; displays o/p = 4  $ expr 2 = 6 ; displays o/p = 0 cause the expression is untrue/false |
| uname -a | Prints kernel name/hostname/kernel release number of the s/m. |
|  | $ uname ; o/p = Linux 🡪 meaning we are running on linux s/m. |
|  | $ host udacity.com ; prints IP address, of site, who handles mail of site etc. |
| bash --version | Gives the version, copyright, license details of bash. |
| uptime | Displays the time machine is running with all details. |
|  | EX: $ uptime; o/p = 10:00:00 up 1 day, 10 min, 2 user, load average: 0.01, 0.1, 1  The format and data could differ between machines. |
| man [command] or  man help [command] | Displays information and all the available usages of a command. |
|  | Ex: $ Man mkdir ; 🡪 Displays usage of mkdir command. |
| history | Records all the commands ran in the s/m (commands used from the beginning). |
|  | $ history; displays all the commands used sequentially. |
| Ctrl + R | To search from entire shell history. |
|  | $ cltr+R; 🡪 displays reverse-i-search, type in the command you were searching it will be displayed. |
| help | Displays usage of all the supported commands of CLI.  Some commands differ in CLI as different versions of Linux/unix supports different command usages. |
|  | $ help history; 🡪 displays the usage of history.  $ man help history 🡪 same as above. |
| tab | Tab key can be used to find out the available files in the system. |
|  | $ cd c 🡪 press Tab ; all the available files starting from “C” will be dislayed. |
| wc xx | Displays how many lines, words, bytes in the file.  Can be used only on file not on folder/dir(directory) |
|  | $ wc file.txt ; o/p 🡪 12 390 330 file.txt .  So there are 12 lines, 390 words of 330 bytes data in file.txt |
| diff fileX fileY | Compares filex with filey and displays all the mismatched lines b/w files.  Can only compare b/w files of same format (i.e., extension)  $ diff File1.txt File2.txt ; |
| apropos | Used in Unix s/m to search in a command’s description. |
|  | Ex: $ apropos edit; all the commands containing edit in their description will be displayed.  Ex: $ apropos copy paste; all commands containing copy / paste in description is shown. |
| start \\IPaddress | To connect to any device (laptop) using it’s IP address and access it’s data. |
| Ping | * To Test a connection request between a host and server. * Keeps running until it is stopped with Ctrl+C   Ex: $ ping 8.8.8.8 🡪 pings google’s DNS server with a connection request time and data stats.  Ex: $ ping –c3 8.8.8.8 🡪 pings google’s DNS server 3 times.  Ex: $ ping yahoo.com 🡪 pings yahoo and returns it’s IP with data msgs. |
| bc | Opens Basic calculator terminal with display of license, copyright information of the software used. Can do all the arithmetic operations.  Ex: $ bc🡪 terminal opens.  Type pi = 10; r=70; a= 100;  Now type pi+r-100 🡪 o/p = -20 |
| vim/ less/ nano/ joe/ emacs | To login and edit the file using terminal (similar to editor, but file content is displayed on terminal). |
|  | EX: $ less e.txt 🡪 opens the file e.txt in terminal, use CLI to edit/cut/rename file and save. |
| df –h OR  free | Prints the memory (total, used, free space) details of the server/home machine. |
| rm [option] file\_name | Removes the file specified with some specifications. |
| [Option] | -r, -R, --recursive: Remove directories and their contents recursively.  -i : Prompt before every removal.  -- version: Display version information, and exit. |
|  | Ex: $ rm –r file4 ; delete all the contents of file4. |
| cp [src] [dest] | Copies file from source to destination.  $ cp C:/user/file backup; copies all files from source to backup folder |
| mv [src] [dest] | To move/rename files from source to destination.  $ mv file1.txt folder4 ; 🡪 moves file file1.txt to folder4;  $ mv file1.txt file9.txt 🡪 renames file1 as file9;  $ mv file.txt file.docx 🡪 changes file type from txt to docx; |
| unzip xxx | To unzip a compressed zip file. |
| unzip –l xxx | To display contents of zip file without unzipping it. |
| shutdown [option] [time] | Option to shut down or restart or reboot the system.  Ex: $ shutdown –h now ; s/m shutdown now  $ shutdown –h +10; s/m shutdown after 10 minutes  $ shutdown –r now; s/m reboot now |
| cat | 1. Displays content of one or more file.   $cat file.txt ; 🡪 displays content of file.txt  $ cat file1.txt file2.html file3.docx ; 🡪 displays content of all 3 files. |
| 1. Copy and paste complete content of one file to another file.   $ cat file1.txt > file2.html ; 🡪 entire data of file2 will be deleted and pasted with file1 data. |
| 1. Create a new file and add contents from old to new file.   $ cat file1.txt > file4.docx; 🡪 Creates a new file file4.docx if not already present then copies data from file1.txt to file4.docx. |
| 1. To append any data into the file   $ cat >> file1.txt ; |
| SSH XXX | To login to remote host |
| FTP XXX | To login to remove FTP location |
| ipconfig [XXX] | To display windows IP configuration |
|  | Ex: $ ipconfig 🡪 displays IPv4 address etc,  $ ipconfig /all 🡪 displays detailed n/w information.  $ ipconfig /renew 🡪 renews all adapters |
| glob | Used for pattern matching of file names. |
| multiple pattern | Suppose files are f1.txt , f2.css, F3 etc.,  $ ls \*.txt 🡪 shows all files ending with .txt  $ ls f1\* 🡪 shows all files starting with f1 |
| single pattern | Suppose files are f1.txt , f2.css, F3 etc.,  $ ls f?.txt 🡪 shows files starting with f, ending with .txt (just ? is mapped) |
| netcat [ nc xxxx] | Netcat is used for any random TCP and UDP connections and listeners.  It can also be used for HTTP connection with the help of $ printf.   * It can open any TCP/UDP connection * Send UDP packets to server or receive from client. * Can listen/connent to any TCP/UDP ports * Deal with both IPv4 and IPv6, do port scanning. * $ Ctrl+D 🡪 command to close the nc Session   Ex: 1. TCP Client – Server Model (between 2 different `machines)  On client (machine 1) 🡪 $ nc –l 1234 : nc is listening on port 1234  On Server 🡪 (machine 2) 🡪 $ nc 127.0.0.1 1234 : nc connects client and server at port 1234 successfully.  Ex: 2. TCP Client – Server Model (between same machine)  Open 2 tabs on same machine and try below commands on the tabs:  $ nc –l 2046 : nc is listening on port 1234  $ nc localhost 2046 : connection established at port 2048 on same machine.  Ex: 3. Using nc with printf to connect to server using HTTP  $ nc printf 'HEAD / HTTP/1.1\r\nHOST: www.google.co.in\r\n\r\n' | nc www.google.co.in 2046   * This helps to connect to Google using HTTP request and HTTP response will be displayed on the terminal.   Refer to manual page for more details ($ man help nc) |
| lsof | Lists all the open files  $ lsof 🡪 displays all the files open |
| sudo | Permits a user to execute a command as superuser/another user.  $ sudo lsof –I 🡪 lists all the open files running on ports. |
| traceroute [domain name] | Used to display route/hops for traffic to reach client from server.  Ex: $ traceroute google.com 🡪 displays no of hops for google to send data back to host or vice versa. |
| tcpdump [options] | Prints out contents of packet transfer on network interface.  Need special privilege to use this tcpdump command.  Ex: $ sudo tcpdump –n host 8.8.8.8;   * Ping query, Shows IP address of host, the packet transfer between host and 8.8.8.8 (which is google’s server), request and reply time etc.,   Ex: $ sudo tcpdump –n port 53;   * DNS Query |
|  |  |